

**ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ ТА ІНФОРМАТИКИ**

«ЗАТВЕРДЖУЮ»

Перший проректор

_____ Леонід Бачурін
«_____» _____ 2024 р.

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ДВС 1.02 ТЕОРІЯ СИНТАКСИЧНОГО АНАЛІЗУ І КОМПІЛЯЦІЇ
(ВБ 1.2 за ОПІ)
(шифр і назва навчальної дисципліни)**

Рівень освіти: перший (бакалаврський)

Спеціальності **121 Інженерія програмного забезпечення**
(шифр і назва спеціальності)

Освітня програма **«Інженерія програмного забезпечення»**
(назва освітньої програми)

Мова навчання: українська

Робоча програма навчальної дисципліни «Теорія синтаксичного аналізу і компіляції» для здобувачів вищої освіти за спеціальністю 121 Інженерія програмного забезпечення

17 грудня 2023 року. – 8 с.

Розробники:

Назарова І.А., к.т.н., доцент кафедри ПМІ, доцент

Робоча програма затверджена на засіданні кафедри прикладної математики та інформатики

Протокол №13 від “27” грудня 2023 р.

В.о. завідувача кафедрою прикладної математики та інформатики

(підпис)

(Маслова Н.О.)
(прізвище та ініціали)

27.12.2023 р.

Схвалено науково-методичною комісією галузі знань 12 Інформаційні технології

Протокол № 1 від “ 15 ” 01 2024 р. Голова

(підпис)

(Башков Є.О.)
прізвище та ініціали)

1. Загальна інформація

| | | |
|---|---|--------|
| Форма навчання | Денна | Заочна |
| Статус | Вибіркова | |
| Обсяг в кредитах ЄКТС | 5 | |
| Обсяг в годинах за навчальним планом, разом: в тому числі: | 150 | |
| лекції: | 32 | 6 |
| практичні заняття: | | |
| лабораторні заняття: | 32 | 6 |
| самостійна робота: | 86 | 138 |
| Форма підсумкового контролю | Екзамен | |
| Дисципліну викладають | Викладач Назарова І.А. https://donntu.edu.ua/kitaer/pmi iryna.nazarova@donntu.edu.ua | |

Передумови для вивчення дисципліни: успішному вивченню дисципліни «Теорія синтаксичного аналізу і компіляції» сприяє попереднє опанування такими дисциплінами, як «Основи алгоритмізації», «Програмування», «Об'єктно-орієнтоване програмування», «Системне програмування», «Дискретні структури та алгоритми».

2. Мета вивчення навчальної дисципліни

Метою викладання навчальної дисципліни «Теорія синтаксичного аналізу і компіляції» є формування системи теоретичних знань і практичних навичок з основ теорії, методів і алгоритмів лексичного, синтаксичного і семантичного аналізу, побудови сучасних інформаційних таблиць у трансляторах, підготовки до генерації і генерації команд об'єктного коду, а також в отриманні практичних навичок у використанні стандартних математичних пакетів для розв'язання наукових і інженерних задач.

Компетентності:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК 2. Здатність застосовувати знання у практичних ситуаціях.

ЗК 6. Здатність вчитися і оволодівати сучасними знаннями.

ЗК 7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ФК-2. Здатність приймати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК-6. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки.

ФК-8. Здатність застосовувати і розвивати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

ФК-15. Здатність до алгоритмічного та логічного мислення.

Програмні результати навчання:

ПРН 1. Вміння аналізувати проблеми щодо створення програмного забезпечення.

ПРН 9. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.

ПРН 7. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.

ПРН 17. Знати, розуміти і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення.

ПРН 19. Знати, розуміти і застосовувати на практиці фундаментальні концепції і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН 22. Аналізувати, оцінювати і вибирати інструментальні та обчислювальні засоби, технології, алгоритмічні і програмні рішення для розв'язання завдань інженерії програмного забезпечення.

ПРН 27. Знати та мати навички реалізації основних алгоритмів та структур даних програмування

3. Очікувані результати навчання

Основними результатами опанування дисципліни «Теорія синтаксичного аналізу і компіляції» є:

– Загальні правила і способи опису синтаксису. Термінальні і нетермінальні символи. Породжуючі граматики. Граматики мов програмування. Класифікацію мов по Хомському. Правила підстановки. Форму Бекуса-Наура, її модифікацію.

– Графічні способи виведення мови: синтаксичні дерева і діаграми. Алгоритми низхідного і висхідного розбору синтаксичного розбору.

– Граматики, що розпізнають, структуру граматик-розпізнавачей. Структуру автоматів. Розпізнавачі для мов з однозначною граматиною.

– Розпізнавачі для конструкцій мов програмування, що описуються автоматними граматиною. Кінцеві автомати-розпізнавачі для конструкцій мов програмування класів 2 і 3 по Хомському. Алгоритми функціонування автоматів. Породжуючі граматики, що допускають побудову кінцевого автомата-розпізнавача.

– Основні синтаксичні форми запису виразів. Представлення виразів при виконанні програм. Алгоритми перекладу виразів в польський інверсний запис. Алгоритми перекладу польського запису в машинні команди.

– Граматики простого автоматного передумання. Відносини передумання і їх використання. Функції передумання. Синтаксичний аналізатор для граматик автоматного передумання.

– Граматики з операторним передуманням. Передумання операторів. Операторні граматики. Алгоритми розбору методом операторного передумання.

– Алгоритми роботи розпізнавача, що використовують передумання операторів. Правила побудови таблиць множин $L(U)$, $R(U)$, $Lt(U)$, $Rt(U)$. Порівняльні характеристики швидкодії і об'єму пам'яті методів простого передумання, що займається і методів, заснованої на стеку з пріоритетами.

– Аналіз теоретичних передумов проектування лексичних аналізаторів, лексичних категорій і показників, структурних схем лексичних аналізаторів, методів. Методи перекодування вихідної програми і здійснення лексичного контролю. Прямі і посередні лексичні аналізатори. Алгоритми сканування.

Внаслідок вивчення курсу студенти повинні вміти:

- виконувати синтаксичний аналіз конструкцій мов програмування, використовуючи графічні способи виведення мови: синтаксичні дерева і діаграми;
- програмувати прямі методи трансляції виразів, використовуючи алгоритми перекладу виразів в польський інверсний запис, використовувати стек з пріоритетами;
- визначати і будувати відносини для граматик простого автоматного передування, використовувати матриці передування, функції передування, допоміжні множини;
- будувати префіксну, інфіксну і постфіксну форми записів виразів по синтаксичному дереву, таблиці операцій і роздільників, що містять пріоритети і ознаки асоціативності операцій. виконувати розбір сентенціональної форми, обробку конфліктів: стратифікацію і розділення;
- передбачати нейтралізацію помилок, здійснювати виправлення орфографічних помилок, семантичних помилок. визначати помилки при низхідному і висхідному розборах;
- будувати таблиці для аналізу вихідного і синтезу об'єктної програм. складати структури таблиць ідентифікаторів, констант, операцій, міток, їх ініціалізація. здійснювати зв'язки таблиць в трансляторах;
- перекодувати вихідну програму, здійснювати лексичний контроль. виділяти і збирати лексеми. будувати прямі і посередні лексичні аналізатори. виключати коментарі. будувати лексичну згортку програми;
- будувати таблиці для аналізу вихідного і синтезу об'єктної програм. складати структури таблиць ідентифікаторів, констант, операцій, міток, їх ініціалізація. здійснювати зв'язки таблиць в трансляторах;
- перекодувати вихідну програму, здійснювати лексичний контроль. виділяти і збирати лексеми. будувати прямі і посередні лексичні аналізатори. виключати коментарі. будувати лексичну згортку програми.

4. Засоби діагностики результатів навчання

Засобами оцінювання та методами демонстрування результатів навчання можуть бути:

- екзамен;
- семестрове індивідуальне завдання;
- лабораторні роботи.

5. Критерії оцінювання результатів навчання

Критерії оцінювання мають формулювати порядок оцінювання під час поточного контролю (за результатами лабораторних занять та виконання індивідуальних завдань) та підсумкового контролю.

| Поточний контроль для очної форми навчання | | | | | | | Поточний контроль | Іспит | Максимальна сума балів |
|--|------|------|------|------|------|-----|-------------------|-------|------------------------|
| Лр.1 | Лр.2 | Лр.3 | Лр.4 | Лр.5 | Лр.6 | ІДЗ | | | |
| 5 | 5 | 5 | 5 | 5 | 5 | 10 | 40 | 60 | 100 |
| 3 | 3 | 3 | 3 | 3 | 3 | 6 | 24 | | |

Відповідність між шкалами встановлюється наступним чином:

| Оцінка | |
|-----------------------|--------------|
| За 100-бальною шкалою | Для екзамену |
| 90-100 | відмінно |

| | |
|-------|--------------|
| 74-89 | добре |
| 60-73 | задовільно |
| 0-59 | незадовільно |

6. Програма навчальної дисципліни

6.1. Основні теми дисципліни

Тема 1. Мета, задачі, зміст курсу. Компілятори, інтерпретатори, асемблери. Основні поняття. Складові частини трансляторів.

Тема 2. Інформаційні таблиці в трансляторах.

Тема 3. Лексичний аналіз. Реалізація функцій лексичного аналізу в трансляторах.

Тема 4. Організація внутрішнього представлення символів. Програмування сканерів.

Тема 5. Методи опису синтаксису мов програмування. Синтаксичний розбір.

Тема 6. Синтаксичний аналіз конструкцій мов програмування.

Тема 7. Граматики, що розпізнають. Структура граматик-розпізнавачей.

Тема 8. Скінченні автомати – розпізнавачі для конструкцій мов програмування, що описуються автоматними граматиками.

Тема 9. Прямі методи трансляції виразів.

Тема 10. Методи інтерпретації виразів.

Тема 11. Граматики простого автоматного передування.

Тема 12. Синтаксичний аналізатор для граматик автоматного передування.

Тема 13. Граматики з операторним передуванням.

Тема 14. Висхідні розпізнавачі, що базуються на методах операторного передування.

Тема 15. Нейтралізація помилок в компіляторах.

Тема 16. Інтерпретатори. Генерація об'єктного коду.

Теми практичних (семінарських) занять

Не передбачено навчальним планом

6.2. Теми лабораторних занять

| № з/п | Назва теми | Кількість годин | |
|--------------|--|-----------------|--------|
| | | Д.ф.н. | З.ф.н. |
| 1 | Теми 1-2. Проектування структурної схеми компілятора. Організація підпорядкованих таблиць в трансляторах і робота з ними. | 4 | 1 |
| 2 | Тема 3-4. Розробка компонент лексичного аналізатора. Конструювання сканерів. | 4 | 1 |
| 3 | Тема 5-6. Опис синтаксису обраної мови програмування. Розробка алгоритму синтаксичного розбору обраних конструкцій мови програмування. | 6 | 1 |
| 4 | Тема 7-8. Побудова граматик, що розпізнають. Розробка синтаксичного аналізатору для конструкцій мов програмування, що описуються автоматними граматиками | 6 | 1 |
| 5 | Тема 9-10. Реалізація прямих методів інтерпретації. | 6 | 1 |
| 6 | Тема 11-14. Проектування СА для мов програмування що використовують методи передування | 6 | 1 |
| Усього годин | | 32 | 6 |

6.3. Самостійна робота

| № з/п | Назва теми | Кількість годин | |
|-------|--|-----------------|------------|
| | | Д.ф.н. | З.ф.н. |
| 1 | Мета, зміст курсу. Компілятори, інтерпретатори, асемблери. | 10 | 20 |
| 2 | Інформаційні таблиці в трансляторах. | 10 | 20 |
| 3 | Лексичний аналіз. Реалізація функцій лексичного аналізу в трансляторах. | 10 | 20 |
| 4 | Організація внутрішнього представлення символів . Програмування сканерів. | 10 | 20 |
| 5 | Опис синтаксису мов програмування. Синтаксичний розбір. Граматики, що розпізнають. | 10 | 20 |
| 6 | Скінченні автомати – розпізнавачі для конструкцій мов програмування. Автоматні граматики.. | 10 | 20 |
| 7 | Прямі методи трансляції виразів. Граматики передумовання – автоматного, операторного. | 10 | 10 |
| 8 | Інтерпретатори. Генерація об'єктного коду. | 16 | 8 |
| | Усього годин | 86 | 138 |

6.4. Індивідуальні та/або групові завдання

Індивідуальне завдання на тему «Проектування та програмна реалізація лексичного та синтаксичного аналізаторів, для заданої категорії мови програмування».

Метою виконання ІЗ є засвоєння теоретичних знань та отримання практичних навичок у розробці програмних додатків, що реалізують, елементи лексичного та синтаксичного аналізаторів для заданої категорії мови програмування. Індивідуальні варіанти для виконання ІЗ вибираються за списком студентів з переліку завдань за методичними вказівками для розрахункових робіт [2].

7. Література

7.1. Основна

1. Aho A., Ullman J., Sethi R., Lam M. Compilers: Principles, Techniques, and Tools. Addison Wesley, 2017. 1040 p.
2. Cooper Keith D., Torczon Linda. Engineering a Compiler, 3-rd Edition (English) (вид. 3 edition). Morgan Kaufmann, 2022. 848p.
3. Advanced Compiler Design and Implementation (English) (вид. 1 edition). Morgan Kaufmann. 1997-08-15. ISBN 9781558603202.
4. Douglas Thain. Introduction to Compilers and Language Design: Second Edition, Independently published, 2020. 247p.

7.2. Допоміжна

1. Формальні мови, граматики та автомати: Навчальний посібник / Гавриленко С.Ю. – Харків: НТУ «ХПІ», 2021. – 133 с.
2. Спекторський І.Я., Статкевич В.М. Формальні мови та автомати . – Київ: КПІ ім. Ігоря Сікорського, 2019. – 167 с.
3. Формальні мови та алгоритмічні: Навчальний посібник / Гаврилків В.М. – Івано-Франківськ: «Голіней», 2023. – 180 с.

4. Homenda W., Pedrycz W. Automata theory and formal languages. – De Gruyter, 2022. – 232 p.
5. Pettorossi A. Automata theory and formal languages: fundamental notions, theorems, and techniques. Springer, 2022.– 288 p.

7.3. Методична

1. Назарова І.А. Дискретний аналіз: навчально-методичний посібник / І.А. Назарова. Донецьк: ДВНЗ “ДонНТУ”, 2012. 277 с. <http://ea.donntu.edu.ua:8080/jspui/handle/123456789/27328>
2. Методичні вказівки і завдання до виконання практичних, розрахункових і самостійних робіт за курсом «Теорія синтаксичного аналізу і компіляції» для студентів спеціальності 121 Інженерія програмного забезпечення всіх форм навчання/ укл. Дмитрієва О.А. – Покровськ: ДонНТУ, 2020. 78 с.

8. Інформаційні ресурси

1. Compilers: principles, techniques and tools. URL:
[https://repository.unikom.ac.id/50738/1/Compilers%20-%20Principles,%20Techniques,%20and%20Tools%20\(2006\).pdf](https://repository.unikom.ac.id/50738/1/Compilers%20-%20Principles,%20Techniques,%20and%20Tools%20(2006).pdf)
2. Advanced compiler design and implementation. Steven Muchnick. URL:
<https://dl.acm.org/doi/10.5555/286076#secabs>